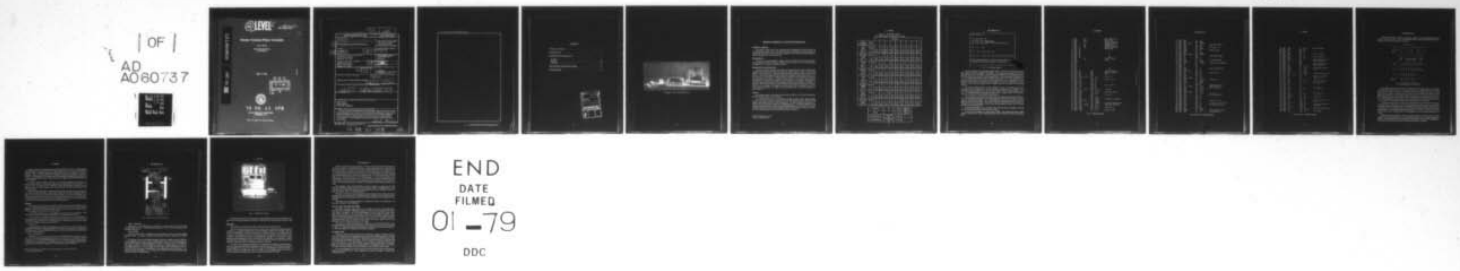
AD-A060 737 NAVAL RESEARCH LAB WASHINGTON D C F/G 17/2
REMOTE TERMINAL PLOTTER CONTROLLER.(U)
MAY 78 J C MOORE
UNCLASSIFIED NRL-8217 SBIE-AD-E000 213 NL



OF
AD
A060737

END
DATE
FILMED
01-79
DDC

(12) NW LEVEL II

ade000213

NRL Report 8217

# Remote Terminal Plotter Controller

John C. Moore

*Radar Techniques Branch*
*Radar Division*

May 17, 1978





NAVAL RESEARCH LABORATORY
Washington, D.C.

(1) Final rept.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NRL Report 8217
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (6) REMOTE TERMINAL PLOTTER CONTROLLER .
5. TYPE OF REPORT & PERIOD COVERED: Final report on one phase of a continuing NRL Problem
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) John C. Moore
8. CONTRACT OR GRANT NUMBER(s): (14) NRL-8217

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Research Laboratory, Washington, D.C. 20375
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61153N RR0310143; R02-96.101

11. CONTROLLING OFFICE NAME AND ADDRESS: (11) 17 Department of the Navy, Office of Naval Research, Arlington, VA 22217
12. REPORT DATE: May [redacted]78
13. NUMBER OF PAGES: 15

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): (12) 15p.
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report): (16) RR03101

Approved for public release; distribution unlimited. (17) RR03101043

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

(18) SBIE (19) AD-E000 213

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Remote plotter
Plotter controller
Microprocessor application

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A remote terminal incremental plotter controller operating within the restrictions of narrow bandwidth telephone lines is described. The device, with minor modifications, will operate nearly all incremental plotters from any computer large enough to hold a software plot package and having a teletypewriter or time-share output. The heart of the device is a National SC/MP microprocessor kit. A patent application covering this device has been submitted under Navy Case 61,576.

251 950

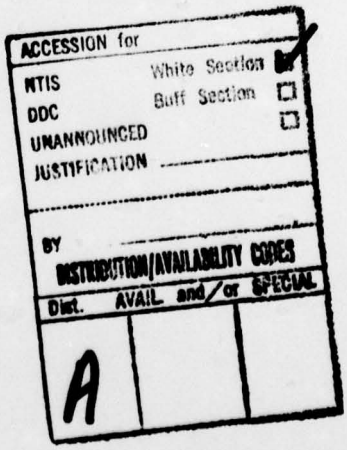
DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

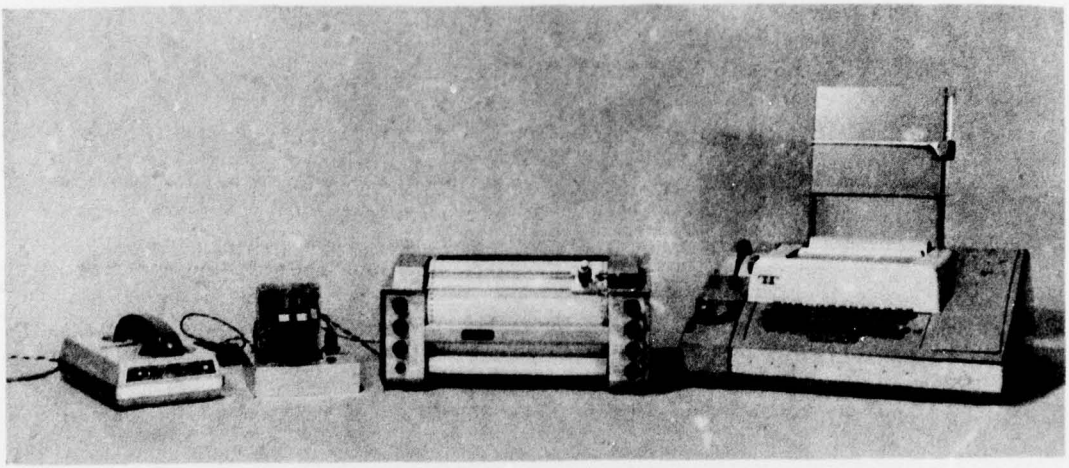


SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS







Frontispiece — *Remote Terminal Plotter System*

# REMOTE TERMINAL PLOTTER CONTROLLER

## GENERAL PURPOSE

This device is used to control an incremental plotter operating within the restrictions of narrow bandwidth telephone lines. The device, with minor modifications, will operate nearly all incremental plotters from any computer large enough to hold a software plotting package and having a teletypewriter or time-share output.

## BACKGROUND

There are no devices available to perform the above function which use existing incremental plotters and computer software. Remote plotters available now are complete packages including plotter, controller, and computer software.

## DESCRIPTION OF OPERATION

The basic method of operation of this device is to decode plotter moves (in direction and magnitude) that have been encoded into ASCII 7-bit codes. Shown in Table 1 are the ASCII characters used in this device. The eight possible plotter movements are shown in the left two columns. Nine possible magnitudes from one plotter increment to nine plotter increments are shown at the top of the table. An ASCII character is assigned to each combination of direction and magnitude. Each character is unique and represents only one direction and magnitude combination. The characters are assigned in such a manner as to encode the direction in the three least significant bits (LSB) and the magnitude in the four most significant bits (MSB). In addition to the 72 ASCII characters needed for direction and magnitude (8 direction × 9 magnitude), four other characters are assigned for pen-up and pen-down commands, and plotter-on and plotter-off commands. An example of the decoding of one of the ASCII characters is shown in Fig. 1.

### Software

The program, stored in the random access memory (RAM) of a National Semiconductor SC/MP microprocessor kit is listed in Fig. 2.

The 256-word RAM starts at location $200_H$* and goes to location $2ED_H$. The remaining locations from $2EE_H$ to $2FF_H$ are reserved for use by the read-only memory (ROM) to store register values during operation of the microprocessor. All address locations and memory contents at those locations are represented by hexadecimal values. The ROM supplied with the SC/MP kit contains programs to allow entering and displaying values in the RAM. The plotter decoding program uses one of these programs (GECO) to read in ASCII characters and send them to the teletypewriter until the first plotting character is detected.

Manuscript submitted March 2, 1973.
*H denotes a hexadecimal value.

Table 1 — PLOTTER CODES
7-bit ASCII CODE: Hexadecimal — (Octal)

| Plotter Commands (Pin Numbers) | Direction | Multiplier 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Drum-Down Code: 01 (Pin 6) | +X,0 0 → | ( | 0 | 8 | @ | H | P | X | \ | h |
| | | 28 (050) | 30 (060) | 38 (070) | 40 (100) | 48 (110) | 50 (120) | 58 (130) | 60 (140) | 68 (150) |
| Drum-Down & Carriage-Left Code: 05 | +X,+Y 1 ↗ | ) | 1 | 9 | A | I | Q | Y | a | i |
| | | 29 (051) | 31 (061) | 39 (071) | 41 (101) | 49 (111) | 51 (121) | 59 (131) | 61 (141) | 69 (151) |
| Carriage-Left Code: 04 (Pin 7) | 0,+Y 2 ↑ | * | 2 | : | B | J | R | Z | b | j |
| | | 2A (052) | 32 (062) | 3A (072) | 42 (102) | 4A (112) | 52 (122) | 5A (132) | 62 (142) | 6A (152) |
| Drum-Up & Carriage-Left Code: 06 | -X,+Y 3 ↖ | + | 3 | ; | C | K | S | [ | c | k |
| | | 2B (053) | 33 (063) | 3B (073) | 43 (103) | 4B (113) | 53 (123) | 5B (133) | 63 (143) | 6B (153) |
| Drum-Up Code: 02 (Pin 5) | -X,0 4 ← | , | 4 | < | D | L | T | \ | d | l |
| | | 2C (054) | 34 (064) | 3C (074) | 44 (104) | 4C (114) | 54 (124) | 5C (134) | 64 (144) | 6C (154) |
| Drum-Up & Carriage-Right Code: 0A | -X,-Y 5 ↙ | - | 5 | = | E | M | U | ] | e | m |
| | | 2D (055) | 35 (065) | 3D (075) | 45 (105) | 4D (115) | 55 (125) | 5D (135) | 65 (145) | 6D (155) |
| Carriage-Right Code: 08 (Pin 8) | 0,-Y 6 ↓ | . | 6 | > | F | N | V | ↑ | f | n |
| | | 2E (056) | 36 (066) | 3E (076) | 46 (106) | 4E (116) | 56 (126) | 5E (136) | 66 (146) | 6E (156) |
| Drum-Down & Carriage-Right Code: 09 | +X,-Y 7 ↘ | / | 7 | ? | G | O | W | ← | q | o |
| | | 2F (057) | 37 (067) | 3F (077) | 47 (107) | 4F (117) | 57 (127) | 5F (137) | 67 (147) | 6F (157) |

| | | | |
|---|---|---|---|
| Pen-Down | & | 26 (046) | CODE: 20 (Pin 12) |
| Pen-Up | ' | 27 (047) | CODE: 10 (Pin 11) |
| TTY-On/Plotter-Off | CTRL-SHIFT M | 1D (035) | |
| TTY-Off/Plotter-On | CTRL-SHIFT N | 1E (036) | |

ASCII Character "S"

7-bit octal code = 123

7-bit binary code = 1 0 1 0, 0 1 1

4 MSB contain magnitude — 3 LSB contain direction

DIR - 011 = $3_{10}$

MULT - 1010 = $10_{10}$ - bias = $6_{10}$

(A Bias of 4 is encoded into each ASCII character multiplier.)

Therefore, the ASCII character "S" is decoded to produce a direction of 3 (Carriage-Left, Drum-Up) and a multiplier of 6 (six increments).

Fig. 1 — ASCII decoding procedure

A brief description of the decoding program follows. The first 24 locations are used for flags, counters, and array storage for the program. The program starts at location $21B_H$. Program GECO, located in the ROM, is called to receive and decode the first ASCII character sent from the modem and to print it out on the teletypewriter. This first character is checked to see if it is the "Plotter-On/TTY-Off" command which is represented by $1E_H$. This character is produced by a Control-Shift-N key stroke. If the first character is not $1E_H$, the program returns to program GECO to receive and decode the next character. This cycle is repeated until a $1E_H$ is received, at which time the program GECO is called a final time to receive the next plotting character.

The second plotting character is then checked to see if it is a Pen-Down command, represented by $26_H$ (ASCII-&), or a Pen-Up command, represented by $27_H$ (ASCII-'). If a Pen-Down command is detected, the value of $20_H$ is loaded in OUT, the plotting direction location. If a Pen-Up command is detected, the value of $10_H$ is loaded in OUT. In both cases a value of $9_H$ is loaded in *I*, the plotting multiplier location.

The location OUT contains the value which will be decoded by the external integrated circuits to provide the proper signal to the incremental plotter. The location I is used to store the value of the plotting multiplier. This is the number of times the value stored in OUT is repeated while driving the plotter. In the case of the Pen-Up and Pen-Down commands, the multiplier is set to the maximum of nine to allow the pen solenoid to react, since it is slower than the *X* and *Y* stepping motors.

If the plotting character is not a Pen-Up or Pen-Down command, the program then strips off the direction and multiplier values as shown in Fig. 1.

At this point the program is at location $26E_H$ (GEN:). The output array ARY is loaded with nine values of zero. Then the direction stored in OUT is loaded in the first I locations of ARY.

```
 1  0200          FLG                        START PLOTTING FLAG
 2  0201          ICHAR                      ASCII INPUT
 3  0202          OUT                        PLOTTING DIRECTION
 4  0203          I                          PLOTTING MULTIPLIER
 5  0204          CNT                        COUNTER FOR ARY
 6  0205          CTR                        COUNTER FOR TTY BITS
 7  0206  01      DIR                        PLOTTING DIRECTION
 8  0207  05                                 OUTPUT CODES
 9  0208  04                                      |
10  0209  06                                      |
11  020A  02                                      |
12  020B  0A                                      |
13  020C  08                                      |
14  020D  09                                      ▼
15  020E          ARY                        PLOTTING OUTPUT
16  020F                                     ARRAY
17  0210                                          |
18  0211                                          |
19  0212                                          |
20  0213                                          |
21  0214                                          |
22  0215                                          |
23  0216                                          ▼
24  0217  08            NOP                PLOTTING PROGRAM
25  0218  08            NOP                STARTS IN LOCATION
26  0219  08            NOP                021B
27  021A  08            NOP
28  021B  C401  START:  LDI   1            SET FLG = 1
29  021D  C8E2          ST    (FLG)
30  021F  37            XPAH  P3           SET P3 = 0185
31  0220  C485          LDI   85
32  0222  33            XPAL  P3
33  0223  C402          LDI   2
34  0225  36            XPAH  P2           SET P2 = 02F0
35  0226  C4F0          LDI   F0
36  0228  32            XPAL  P2
37  0229  3F    READ:   XPPC  P3           CALL GECO
38  022A  C8D6          ST    (ICHAR)
39  022C  E41E          XRI   1E           CHECK FOR "TTY OFF"
40  022E  9C07          JNZ   $NEXT
41  0230  C400          LDI   0            SET FLG = 0
42  0232  08            NOP
43  0233  C8CC          ST    (FLG)
44  0235  90F2          JMP   $READ        GET NEXT TTY CHARACTER
45  0237  C0C8  NEXT:   LD    (FLG)
46  0239  9CEE          JNZ   $READ
47  023B  C0C5          LD    (ICHAR)      LOAD FIRST PLOTTING CHAR
48  023D  E426  DECODE: XRI   26           CHECK FOR "PEN DOWN"
49  023F  9C0A          JNZ   $TEST1
50  0241  C420          LDI   20           P/DN CODE = 20
51  0243  C8BE          ST    (OUT)
52  0245  C409          LDI   9            REPEAT 9 TIMES
```

**Fig. 2 — SC/MP program listing**

```
 53  0247  C8BB          ST    (I)
 54  0249  9023          JMP   $GEN
 55  024B  C0B5  TEST1:  LD    (ICHAR)
 56  024D  E427          XRI   27          CHECK FOR "PEN UP"
 57  024F  9C0A          JNZ   $TEST2
 58  0251  C410          LDI   10          P/UP CODE = 10
 59  0253  C8AE          ST    (OUT)
 60  0255  C409          LDI   9           REPEAT 9 TIMES
 61  0257  C8AB          ST    (I)
 62  0259  9013          JMP   $GEN
 63  025B  C0A5  TEST2:  LD    (ICHAR)
 64  025D  D407          ANI   7           STRIP OFF DIRECTION
 65  025F  31            XPAL  P1
 66  0260  03            SCL
 67  0261  C106          LD    6(P1)       GET DIRECTION CODE
 68  0263  C89E          ST    (OUT)
 69  0265  C09B          LD    (ICHAR)     RELOAD PLOTTING CHARACTER
 70  0267  1C            SR
 71  0268  1C            SR
 72  0269  1C            SR
 73  026A  FC05          CAI   5           STRIP OFF MULTIPLIER
 74  026C  C896          ST    (I)
 75  026E  A894  GEN:    ILD   (I)
 76  0270  C40E          LDI   0E          ADDRESS OF ARY
 77  0272  31            XPAL  P1
 78  0273  C409          LDI   9
 79  0275  C88E          ST    (CNT)
 80  0277  C400  LOOP1:  LDI   0
 81  0279  CD01          ST    @+1(P1)     ZERO-OUT ARY
 82  027B  B888          DLD   (CNT)
 83  027D  9CF8          JNZ   $LOOP1
 84  027F  C40E          LDI   0E
 85  0281  31            XPAL  P1          ADDRESS OF ARY IN P1
 86  0282  C07F  LOOP2:  LD    (OUT)       STORE OUT INTO ARY
 87  0284  CD01          ST    @+1(P1)     FOR (I) VALUES
 88  0286  B87C          DLD   (I)
 89  0288  9CF8          JNZ   $LOOP2
 90  028A  C408  PLOT:   LDI   8
 91  028C  C878          ST    (CTR)       SET CTR = 8
 92  028E  06    2:      CSA               WAIT FOR START BIT
 93  028F  D420          ANI   20
 94  0291  9CFB          JNZ   $2
 95  0293  C457          LDI   57
 96  0295  8F04          DLY   4           DELAY ½ BIT TIME
 97  0297  06            CSA
 98  0298  D420          ANI   20          IS START BIT STILL THERE?
 99  029A  9CF2          JNZ   $2          NO! GO BACK AND WAIT
100  029C  C400          LDI   0
101  029E  36            XPAH  P2
102  029F  C06E          LD    (ARY)
103  02A1  CA00          ST    0(P2)       PLOT FIRST VALUE OF ARY
104  02A3  C400          LDI   0           BY ATTEMPTING TO
```

**Fig. 2 (continued) — SC/MP program listing**

```
105  02A5  CA00          ST    0(P2)        WRITE INTO ROM
106  02A7  C40F          LDI   0F
107  02A9  31            XPAL  P1
108  02AA  C47E          LDI   7E
109  02AC  8F08          DLY   8            DELAY 1 BIT TIME
110  02AE  06            CSA
111  02AF  D420          ANI   20           GET BIT (SENSE B)
112  02B1  9802          JZ    $3
113  02B3  C401          LDI   1            SAVE BIT VALUE (0 OR 1)
114  02B5  1F      3:    RRL                ROTATE INTO LINK
115  02B6  01            XAE
116  02B7  1D            SRL                SHIFT INTO CHARACTER
117  02B8  01            XAE                RETURN CHAR TO E
118  02B9  C501          LD    @+1(P1)
119  02BB  CA00          ST    0(P2)        PLOT REMAINING VALUES
120  02BD  C400          LDI   0            OF ARY
121  02BF  CA00          ST    0(P2)
122  02C1  B843          DLD   (CTR)
123  02C3  9CE5          JNZ   $LOOP3       RETURN FOR NEXT TTY BIT
124  02C5  40      CTEST: LDE
125  02C6  D47F          ANI   7F           MASK INPUT CHARACTER
126  02C8  01            XAE                STORE IT IN E
127  02C9  40            LDE
128  02CA  C836          ST    (ICHAR)      STORE IT IN ICHAR
129  02CC  E41D          XRI   1D           IS ICHAR = P/OFF?
130  02CE  9814          JZ    $5           YES!  GO TO 5
131  02D0  40            LDE
132  02D1  03            SCL
133  02D2  FC70          CAI   70           IS ICHAR ≥ 70?
134  02D4  9406          JP    $4           YES!  GO TO 4
135  02D6  40            LDE
136  02D7  03            SCL
137  02D8  FC26          CAI   26           IS ICHAR ≥ 26?
138  02DA  940C          JP    $6           YES!  GO TO 6
139  02DC  C400    4:    LDI   0
140  02DE  C823          ST    (OUT)        SET OUT = 0
141  02E0  C46D          LDI   6D
142  02E2  31            XPAL  P1           ADDRESS OF GEN IN P1
143  02E3  3D            XPPC  P1           CALL GEN
144  02E4  C41A    5:    LDI   1A
145  02E6  31            XPAL  P1           ADDRESS OF START IN P1
146  02E7  3D            XPPC  P1           CALL START
147  02E8  C43C    6:    LDI   3C
148  02EA  31            XPAL  P1           ADDRESS OF DECODE IN P1
149  02EB  C015          LD    (ICHAR)
150  02ED  3D            XPPC  P1           CALL DECODE
```

Fig. 2 (continued) — SC/MP program listing

Location $28A_H$ (PLOT:) begins the actual data output to the external integrated circuits and from there to the plotter. This is accomplished while the next ASCII plotting character is being received by the microprocessor (see Fig. 3).

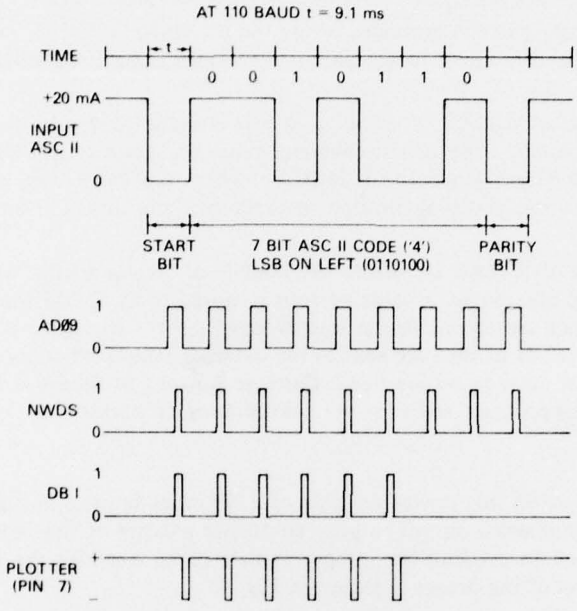



Fig. 3 — Timing diagram showing output to pin 7 of plotter during input of 7-bit ASCII code

The program waits for the start bit to be received from the modem. When it is received, the program waits for 1/2 bit time. If the start bit is not there, the program returns to waiting for the start bit. When it has been received for 1/2 bit time, the first value stored in ARY is fetched and an attempt is made to write it in location $O_H$ of the ROM. This sets bit 9 of the address bus and places the value stored in ARY on the data bus. Since data cannot be written into the ROM, the data is not actually written into memory, but is available on the data bus. The external integrated circuits detect the value on the data bus and the fact that address bit 9 is set. A pulse is then generated and sent to the appropriate input on the plotter.

The program then waits for one full bit time and fetches the first bit of the ASCII character being received. After this bit is detected, the second value stored in ARY is again fetched and again an attempt is made to write it to the ROM with the result of another signal being sent to the plotter. This cycle repeats until all seven bits of ASCII and the parity bit have been fetched and sent to the external integrated circuits and on to the plotter. As the ASCII bits are received they are not "echoed" back to the teletypewriter, so the incoming character is not printed.

When the plotting multiplier stored in *I* is less than nine, the remaining locations in ARY contain the previously stored zeroes. As the appropriate ASCII bit is received and these zero values of ARY are fetched, they are also placed on the data bus, but the external integrated circuits send no output pulses to the plotter and no pen movement is actuated.

After the parity bit is received and the last value of ARY is sent out, the newly received ASCII character is stored in ICHAR. Its value is tested to see if it is $1D_H$ (ASCII-Control-Shift-M). This is the character to terminate plotting and restore the teletypewriter to the printing mode. When this Plotter-Off/TTY-On command is received, the program is returned to START: at the beginning of the program, where the incoming ASCII characters will be printed on the teletypewriter and not sent to the plotter until once again the Plotter-On/TTY-Off command is received.

If the new value in ICHAR is not $1D_H$, a test is made to determine the validity of the incoming plotter characters. The ASCII character must be between $26_H$ and $6F_H$ inclusive. If this test is true, the program returns to DECODE: where the Pen-Down and Pen-Up tests are made, and then on to the decoding routine to determine the direction and magnitude of the plotting command.

When the value of ICHAR is outside the bounds of $26_H$ and $6F_H$, with the exception of $1D_H$ which was tested previously, a value of zero is stored in OUT and the program returns to GEN: where all zeroes are stored in the output array ARY. As the next ASCII character is received, these nine values of zero are sent to the external integrated circuits. Therefore, when a nonplotting character such as a Line Feed, Carriage Return, or others is received, the plotter is not activated and the program waits for the next plotting character.

**Hardware**

The hardware used in this device consists of a National Semiconductor SC/MP microprocessor kit, an input transistor circuit to provide proper polarity of the input signals, and four integrated circuits used to produce the proper signals to be used by the incremental plotter. The basic configuration of the device is shown in Fig. 4.

The SC/MP microprocessor kit was assembled as described in the *Users Manual**. One input and four outputs were used in the SC/MP Kit. They are described as follows.

*Input*: The output of a modem is connected to the TTY-IN (+) pin of the SC/MP Kit. In the working model, an Anderson-Jacobson model ADAC 242 modem is used. Since the output of this modem is a positive 20 mA, a 2N2222 transistor is added in series to provide the proper input to the SC/MP Kit, as shown in detail in Fig. 4.

*Output*: TTY

A standard ASR 33 teletype® is connected to the TTY-OUT (+ and -) terminals of the SC/MP Kit as described in the *Users Manual*. The output of the teletypewriter is connected to the modem input. Thus, the SC/MP Kit is wired in a series with the output of the modem and the input of the teletypewriter.

*Output*: DATA BUS

Data bits 0, 1, 2, 3, 4, and 5 of the data bus of the SC/MP Kit are connected to A, B, C, and D of one DM 74LS168 integrated circuit and A and B of the second DM 74LS168 integrated circuit. The DM 74LS168 integrated circuit is a synchronous 4-bit up/down counter. In this device it is being used as a 4-bit register with the counter disabled by grounding the enable pin ($\overline{P}$) and letting the enable pin ($\overline{T}$) and the up/down pin ($U/\overline{D}$) float.

*National Semiconductor Pub. No. 4200113A, National Semiconductor Corp., Santa Clara, CA 95051

®Teletype Corporation Trademark

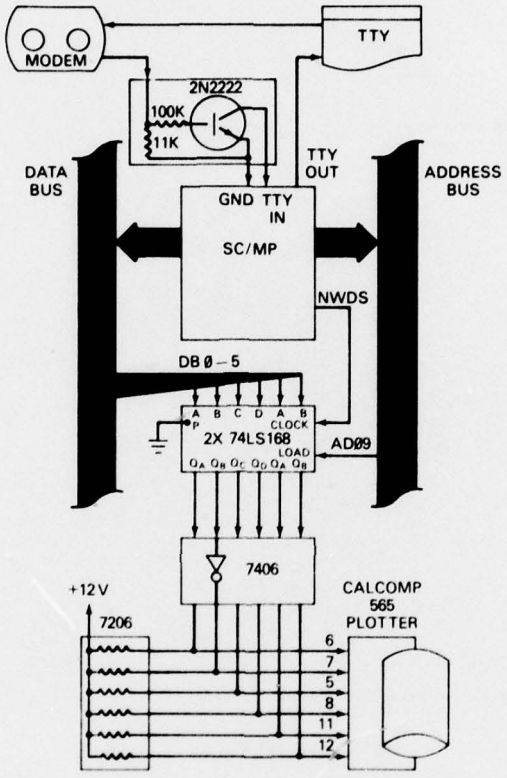



Fig. 4 — Remote Terminal Plotter block diagram

*Output:* Address Bus

Address bit 9 of the address bus is connected to the load input of both of the DM 74LS168 integrated circuits. This allows the data inputs at A, B, C, and D to be loaded when the address bit 9 is set.

*Output:* NWDS

The Write Data strobe is connected to the clock input of both of the DM 74LS168 integrated circuits. This output is strobed every time data is written to memory in the SC/MP Kit.

The outputs of the two DM 74LS168 integrated circuits are buffered by an SN 7406 Hex inverter buffer/driver with open-collector and high-voltage output. The two DM 74LS168 and one SN 7406 integrated circuits are supplied by the same +5-V source used by the SC/MP Kit. The outputs of each of six SN 7406 buffers are connected to negative input pins of the CalComp Model 565 plotter. The pin numbers in Fig. 4 refer to the pin connections on the CalComp Model 565 plotter and are documented in the instruction manual provided with it. To properly pull down the buffers, a +12-V supply is also connected to the plotter inputs through resistors contained in a 7206 flat pack.

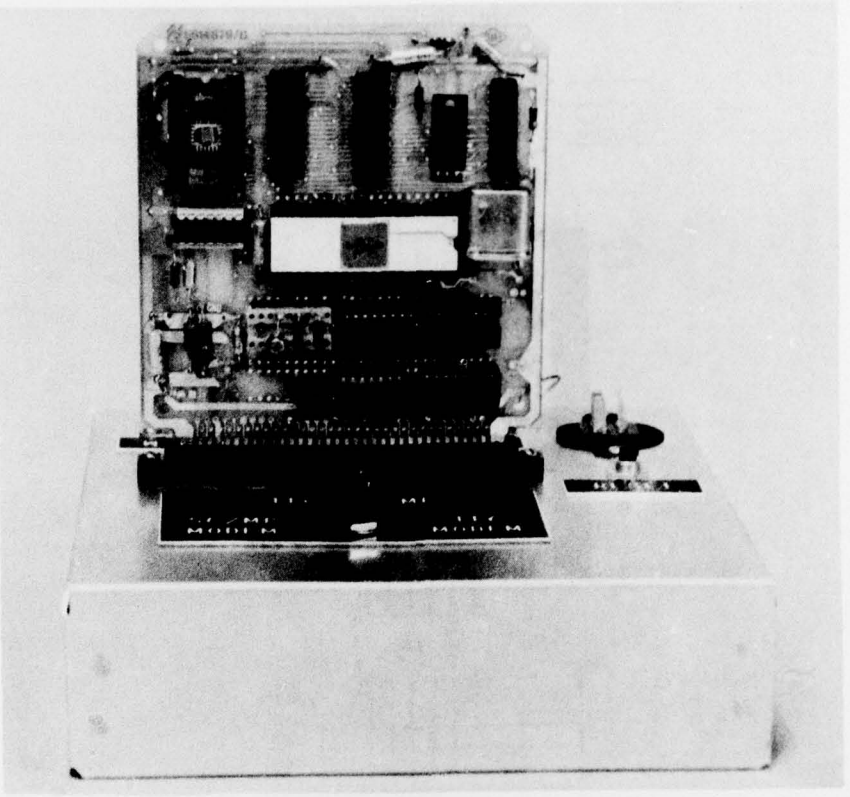


Fig. 5 — SC/MP Plotter Controller

The input transistor circuit and the four output integrated circuits are all mounted on the printed circuit board containing the SC/MP Kit, making the very compact device shown in Fig. 5.

**Data Flow**

The operation of the plotter controller is described as follows (see Fig. 4):

With the controller connected as shown in Fig. 4 and a suitable time-sharing computer connected to a standard telephone line, the phone handset is placed on the modem. The output of the teletypewriter is connected directly to the modem, which in turn is connected to the computer. The teletypewriter can now communicate with the computer. The computer output received by the modem is entered into the microprocessor through its TTY input lines. For the modem and microprocessor kit used to demonstrate this controller, it was necessary to build a switching device (the 2N2222 transistor) to convert the positive 20-mA current output of the modem to a switch closure for the microprocessor TTY input line.

The input TTY command received by the microprocessor is decoded as described previously. Until the special Plotter-On command is received, the input TTY characters are sent to the teletypewriter by way of the TTY-Out line of the microprocessor and the character is printed. Thus the controller acts only as a buffer in the computer output line, and the modem-teletypewriter combination acts as a typical remote computer terminal.

When the special Plotter-On command is received, the software in the microprocessor kit decodes this character, as described previously. As each plotter character is now received by the microprocessor, it is decoded into the proper command needed by the incremental plotter. This plotter command is placed on the data bus of the microprocessor. Also, address bit 9 is set on the address bus of the microprocessor and is sent to the load input of the two integrated circuits (DM 74LS168) used to drive the incremental plotter. The Write Data strobe is set each time a word is written into memory of the microprocessor kit. This strobe is available at the NWDS output of the microprocessor and is used to clock in the plotter command which has been placed on the data bus, but only if the load signal is set by address bit 9 of the microprocessor. Thus the plotter command is entered into the integrated circuits each time address bit 9 is set.

The integrated circuits (DM 74LS168) are used as registers to control the SN 7406 buffer/driver which is connected directly to the incremental plotter. This buffer/driver is used to convert the 5-V logic level used by the microprocessor kit and the 74LS168's to the 12-V logic level needed to drive the incremental plotter.

Therefore, the decoded input command from the microprocessor kit places the proper voltage level on the input pins of the incremental plotter, causing it to move the plotter pen in the proper direction corresponding to the plottor command received by the controller from the computer.

The device and its associated equipment, including the modem, the teletypewriter, and the plotter, are shown in the frontispiece.

## ADVANTAGES AND NEW FEATURES

The biggest advantage of this system is in the savings to the user. Conventional plotter controllers cost a minimum of $600, and usually do not allow remote operation of the plotter. This system is inexpensive. The SC/MP microprocessor costs less than $100. Other advantages include the ability to use any available software plotting package by adding a single software subroutine to generate the ASCII plotter code. Also, any computer with teletypewriter output or time-sharing capability may be used to operate the plotter. In addition, nearly all incremental plotters may be interfaced to this system with minor changes in the output integrated circuits to produce the proper control voltages.

The ability of a single ASCII character to produce up to nine plotter increments in any one direction greatly increases the plotter speed. This feature can be easily set up to operate the plotter at its maximum speed when the software is being operated at maximum efficiency, i.e., when line lengths are in multiples of nine plotter increments.

## ALTERNATIVES

This device could be constructed from any microprocessor capable of computing teletypewriter ASCII characters and having enough memory to store the decoding program. As already stated, any computer with the capability of teletypewriter output and enough memory to hold any software plotting package may be used. Also, most incremental plotters may be interfaced to the microprocessor. If the microprocessor input is capable of operating at different baud rates, the output of the device can be made to approach the maximum speed of the plotter. With more memory associated with microprocessor, the multiplication factor of nine plotter increments per ASCII character could be increased to further increase the plotter speed.

The output integrated circuits used in this device were tailored to operate the CalComp Model 565 incremental plotter. These integrated circuits could be changed to operate other incremental plotters.